**GIT**

1. **Instal GIT**
2. **About GIT**

* Git là gì?
* Một hệ thống quản lý phiên bản phân tán => quản lý source code trên Linux
* Lưu trạng thái của file (project) dưới dạng lịch sử
* Repository là gì?

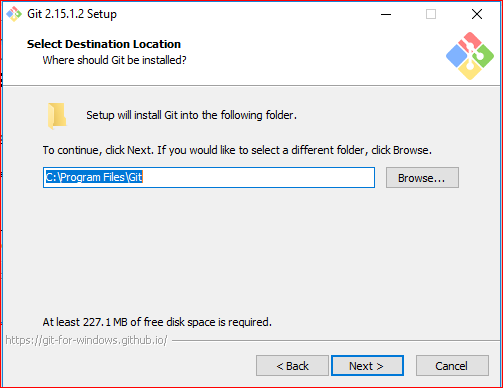


* Remote repository
* Local respository

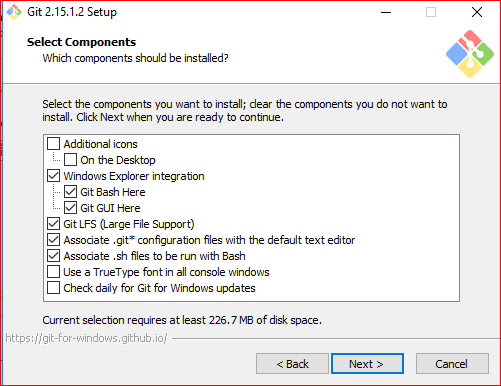
1. **Download GIT for Windows**: <https://git-scm.com/>
2. **Install GIT**



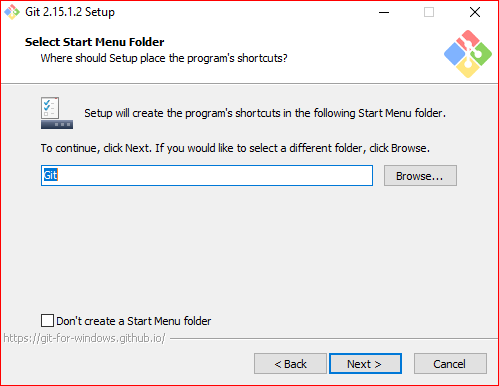
Nhấn “Next”



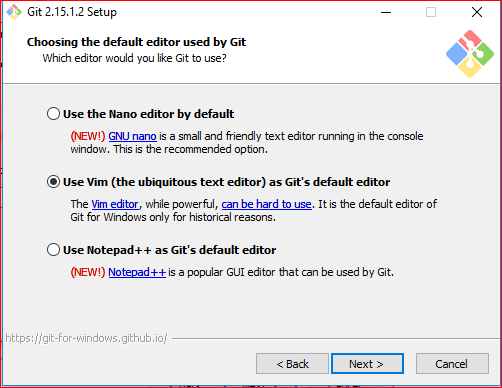
Nhấn “Next”



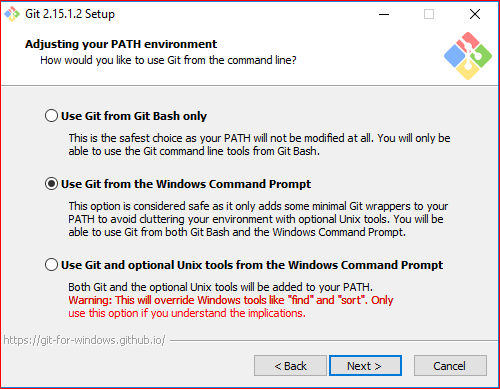
Nhấn “Next”



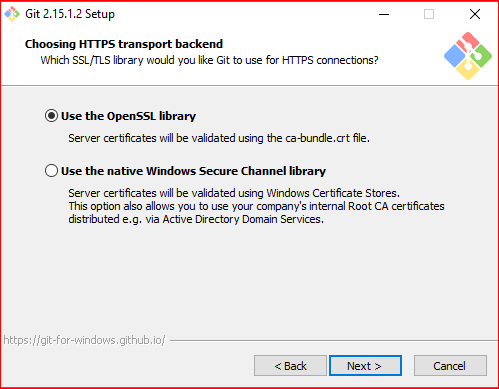
Nhấn “Next”



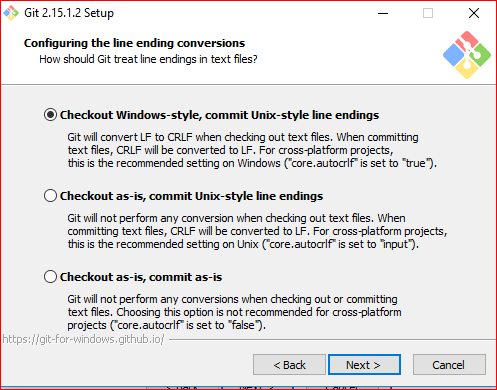
Nhấn “Next”



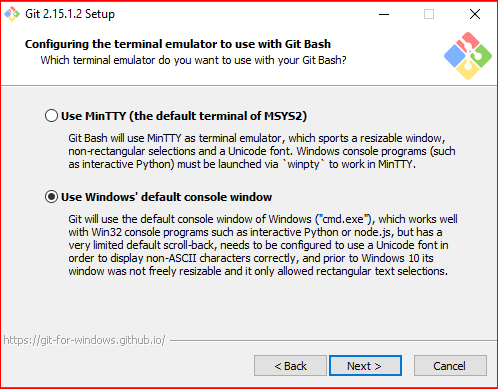
Nhấn “Next”



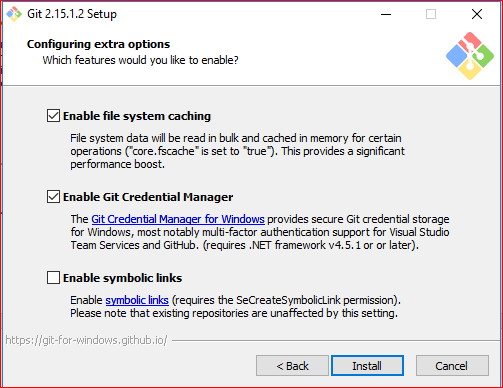
Nhấn “Next”



Nhấn “Next”



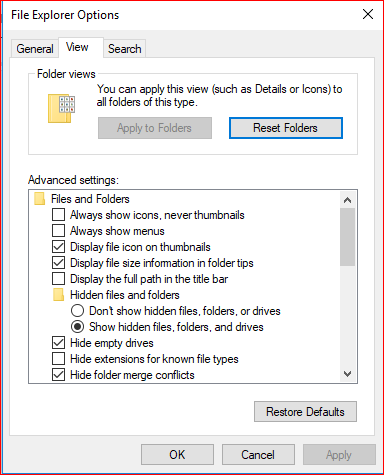
Chọn : Use Windows’ default console window => Nhấn “Next”



Nhấn “Next” => Finish

1. **Using Git**
2. **Change view in forder options :**

* Control Panel => Appearance and Personalization => File Explorer Options => tab View =>



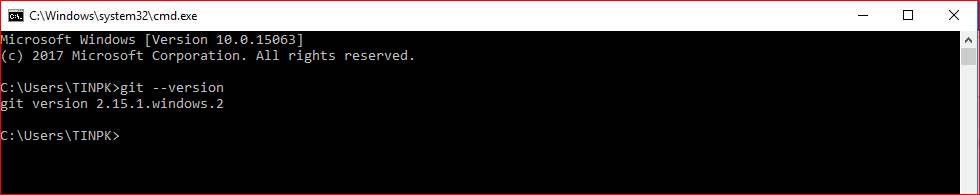
Tick chọn: Show hidden file, folders and drives

Bỏ chọn: Hide extensions for know file types

=> Mục đích : có thể nhìn thấy file .git

1. **Checking git version :**

* **Vào Run => cmd**
* git –version

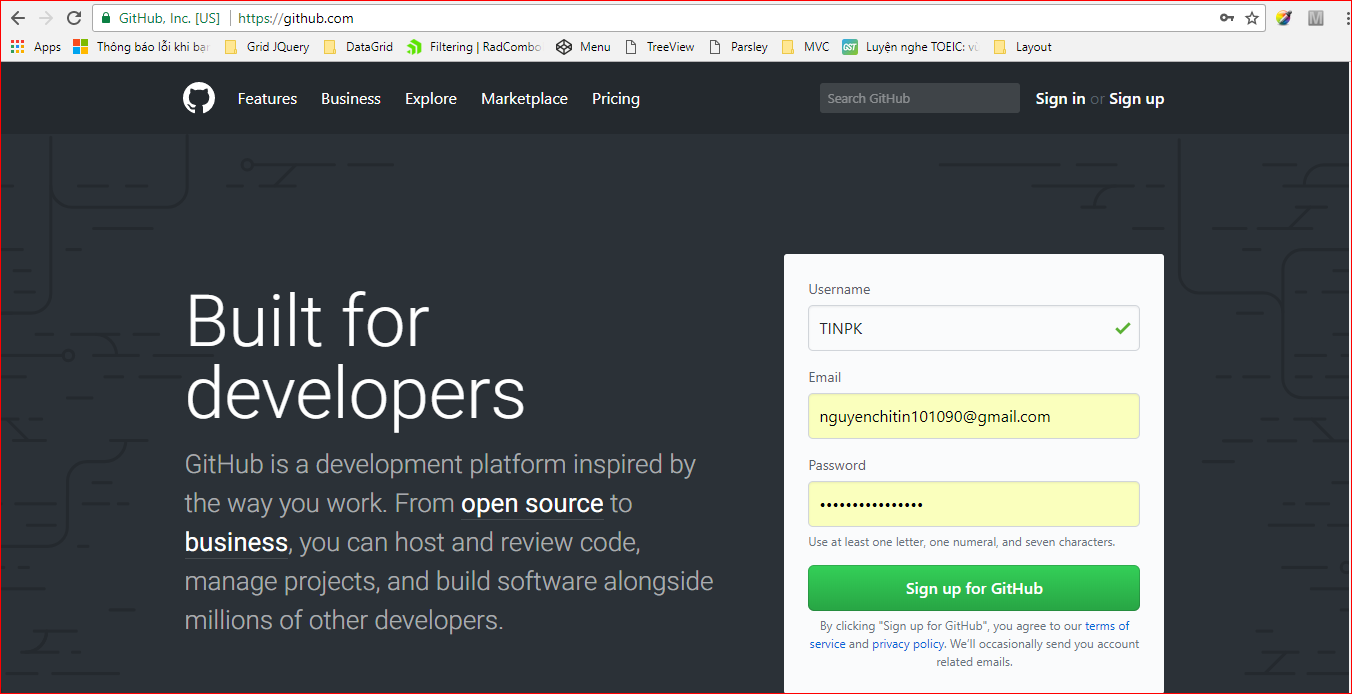


1. **Configured git first:**

* git config
* git config --help
* git help config

1. **Sign up account Github :**

* <https://github.com/>



1. **Configured git synchronize with Github (Same information with Github)**

* git config --global user.name = “TINPK” (để ý dấu cách nữa nhé)
* git config --global user.email = “[nguyenchitin101090@gmail.com](mailto:nguyenchitin101090@gmail.com)”

1. **Test on a project**

* Tạo một project (Folder) GitTest và (Folder) gitproject bên trong (Folder) GitTest.
* Bên trong gitproject tạo file “Test.txt” và “TestSecond.txt”

1. **Initialize Repository (Local repository)**

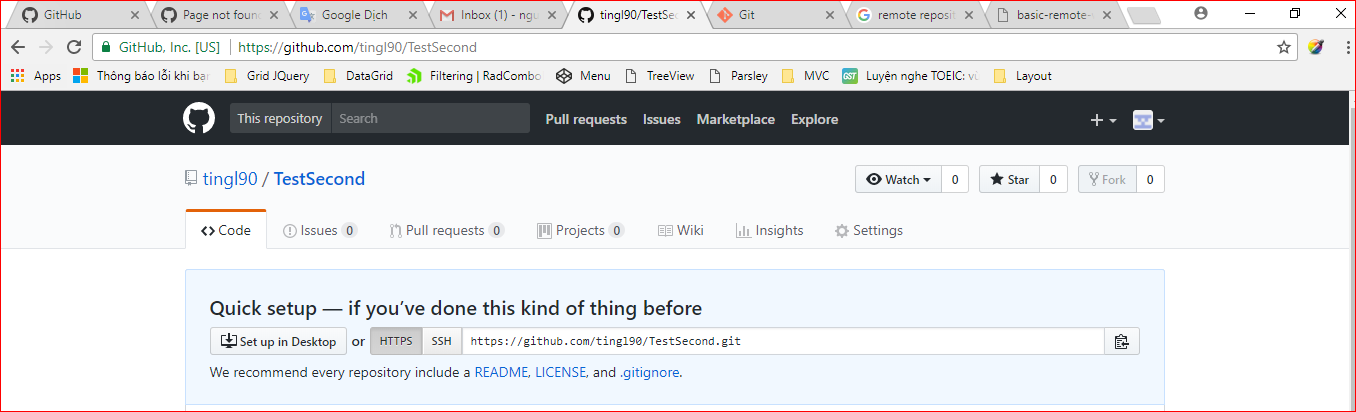
* Vào folder gitproject => Nhấp chuột phải chọn “Git Bash Here”
* **Tạo repository**: git init
* **Xem trạng thái**: git status
* **Kiểm tra số dòng code đã thay đổi :** git diff --stat
* **Xác nhận 1 file (Tracked file)** : git add <tên file> VD: git add “TestSecond.txt”
* **Xác nhận nhiều file** : git add . (Lưu ý dấu chấm nhé)

1. **Upload file on Local Repository**

* git commit -m “note” *(note : ghi bất kỳ)*
* git commit -a -m “note” *(gom 2 lệnh: git add . và git commit -m “note”)*

1. **Create Repository on Github** <https://github.com/>

* Click vào “New repository”
* Đặt tên Repository => Create repository



* Copy dường dẫn :

1. **Push project (Folder) on Github**

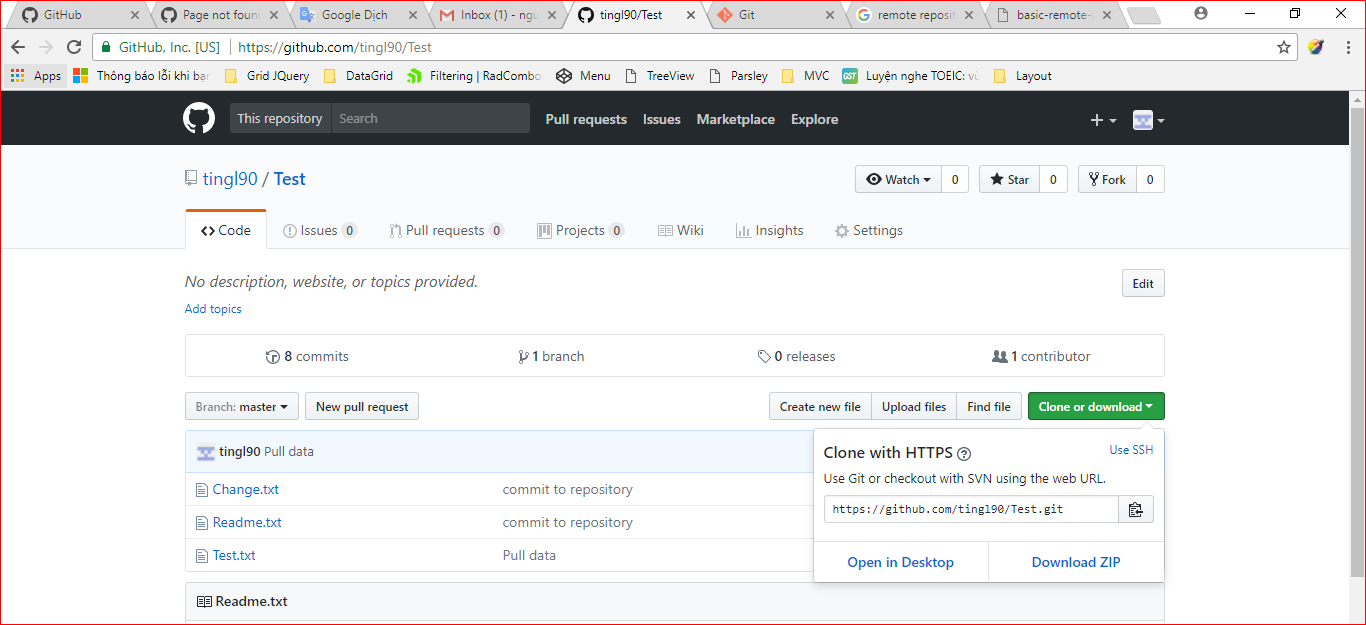
* git remote add <ten project> <url>

VD: git remote add gitproject https://github.com/tingl90/TestSecond.git

* git push <ten project> *(git push --set-upstream <ten project> master)*

VD: git push gitproject (git push --set-upstream gitproject master )

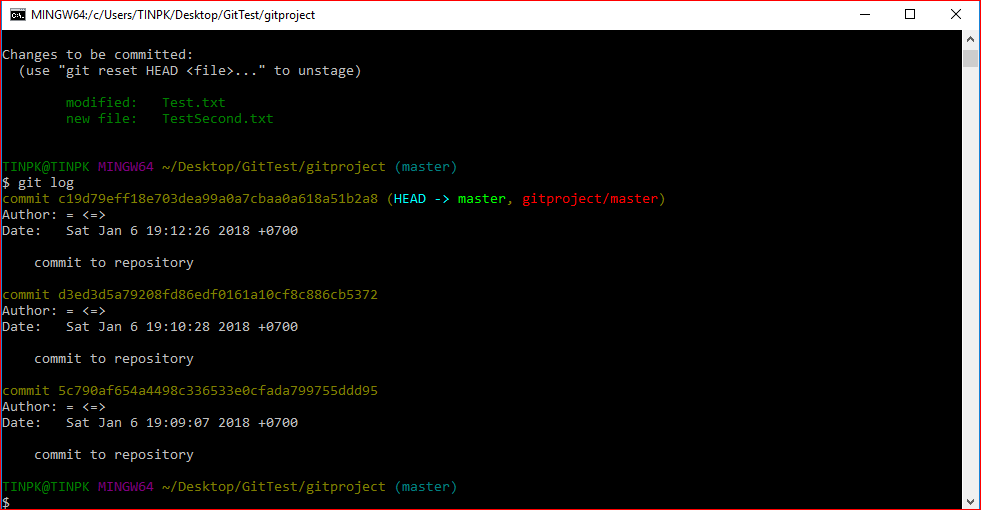
1. **Download file from Github (Backup)**



* Copy đường dẫn
* Vào thư mục cần Paste => Click chuột phải chọn “Git Bash Here”
* git clone <url> vd: git clone <https://github.com/tingl90/Test.git>

1. **View history commit and take file again (Change each commit)**

* git log (Xem lịch sử)



* Copy ID của lần muốn lấy: VD: c19d79eff18e703dea99a0a7cbaa0a618a51b2a8
* git revert <ID vừa copy> (Lấy lại dữ liệu tại thời điểm trước đó)

VD: git revert c19d79eff18e703dea99a0a7cbaa0a618a51b2a8

* Bấm Ctrl + C để dừng
* Shift : q

1. **Pull file back your computer**

* git pull <url>

1. **Working with Branch (Branch)**

* git branch (Xem các nhánh hiện tại)
* git branch <ten branch> (Tạo một nhánh mới)
* git checkout<ten branch> (Chuyển nhánh)
* git checkout -b <ten branch> (Tạo và chuyện nhánh cùng lúc)
* git push --set-upstream <ten project> <ten branch> (Đẩy file khi chưa có branch trên github)
* git push <ten project> <ten branch> (Đẩy file khi chưa có branch trên github)
* git push (Trường hợp branch cũ)
* git show <ID commit> (Duyệt lại các thay đổi trước khi chuyển lên)
* git branch -d <ten branch> (Xóa tại checkout master nhé)
* git branch -D <ten branch> (is not fully merge)
* git push <remote name> --delete <ten branch>

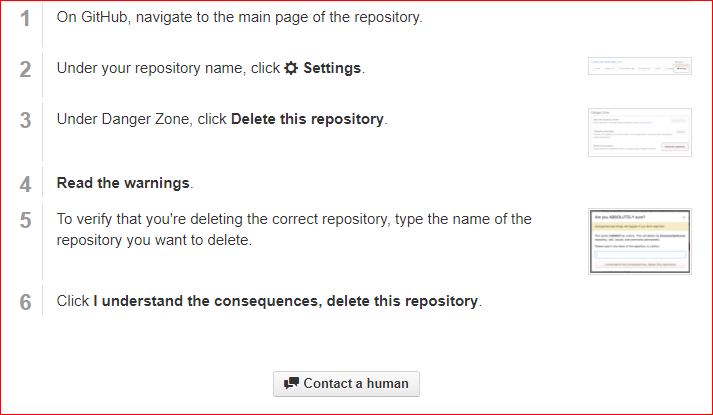
or git push <remote name> :<ten branch> (remote name == url)

VD: git push https://github.com/tingl90/TESTGIT.git --delete master\_dung

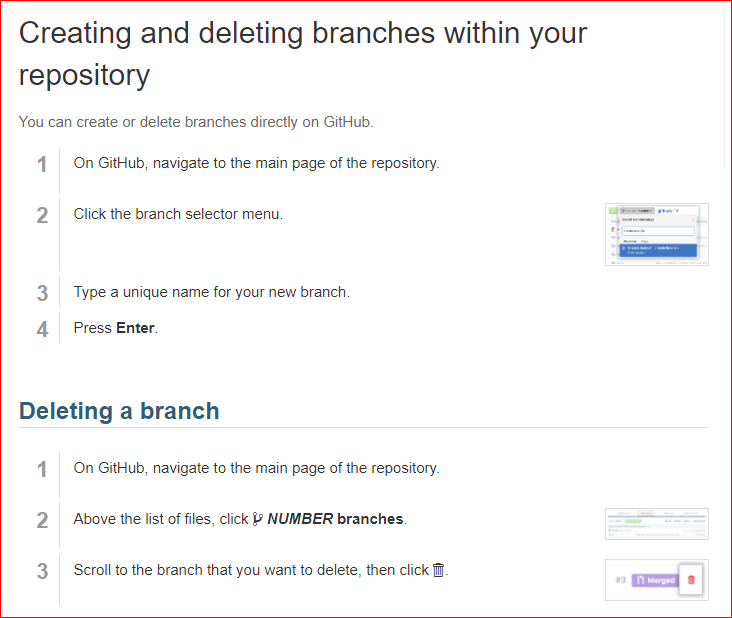
Or git push https://github.com/tingl90/TESTGIT.git :master\_tin

If this is your first push. change it like this! (Issue a forced push with the command)

* git push -f <ten project> <ten branch>
* git branch --v (See the last commit on each branch)
* git branch --merged (See branches are already merged into the branch you are on)
* git branch --no-merged ( See branches that containwork you have not yet merged)



*Delete repository*



*Create and Delete branch*

1. **Merge and Conlicts in GIT**

**Merge**

* git merge <ten branch>

**\*** fatal: the current branch master has no upstream branch. To push the current branch and set the remote as upstream, use:

* git push --set-upstream <ten project> master
* git push <ten project>

**Conlicts**

* Download cài đặt DiffMerge
* git config --global merge.tool diffmerge (Cấu hình Diffmerge)
* git merge <ten branch>
* git mergetool

1. **View list remote branches of Project**

* git ls-remote <ten project>
* git checkout --track -b <ten branch> Checkout remote branch về local (Tracking branches)
* git branch –vv (See what tracking branches you have set up)

1. **Working with Tag**

**Lightweight Tag:** Chỉ đánh dấu snapshot của commit (tham khảo sau)

**Annotated Tag:** Có thể đặt tiêu đề cho tag, Xem được thông tin về người tag, ngày tag

* git tag -a <ten tag> -m “Tên tiêu đề” ( Mặc định là lần commit cuối cùng)
* git show <ten tag> (Xem thông tin commit được gán tag. Mặc định là lần commit cuối )

**Thêm tag cho các commit cũ**

* git log --pretty=oneline (Xem mã checksum của lần commit trước đó)



* git tag -a v0.0 <mã checksum> -m “Tên tiêu đề”

VD: git tag -a v0.0 6904d -m “Tag for Initial commit”

* git push --tags (Đẩy toàn bộ tag lên remote repository)
* git checkout -b <ten branch> <ten tag> (Đưa tag vào một branch)

**$ git push --set-upstream TESTGIT master *(Branch 'master' set up to track remote branch 'master' from 'TESTGIT'.)***

1. **sdfdsf**
2. **fdfdf**

**Sumary:**

- **Kiểm tra version**: git --version

- **Xem thông tin** : git config

- **Trợ giúp**: git config --help

- **Cấu hình username** : git config --global user.name = “tingl90”

- **Cấu hình email** : git config --global user.email = “nguyenchitin101090@gmail.com”

- **Khởi tạo repository**: git init

- **Xem trạng thái**: git status

- **Xác nhận 1 file** : git add “text.txt” **hoặc nhiều file** : git add .

- **Up file len Local repository**: git commit –m “note”

- **Xác nhận và Up file** : git commit -a -m “note”

- **Đẩy project lên github**: git remote add <ten project> <url> => git push <ten project>

: git push --set-upstream <ten project> master (gán master cho project)

: git push origin <ten project>

- **Download file từ Github**: git clone <url>

- **Xem lịch sử**: git log

- **Lấy lại dữ liệu tại thời điểm trước đó**: git revert <ID commit>

**- Kéo file về máy tính**: git pull

**- Tạo một nhánh mới**: git branch <ten branch>

**- Chuyển nhánh**: git checkout <ten branch>

**- Tạo và chuyển sang một nhánh** : git checkout -b <ten branch>

**- Xem tất cả các nhánh**: git branch

**- Đẩy file lên nhánh** : git push --set-upstream origin <ten branch> ( Tạo khi chưa có branch trên github)

- **Duyệt lại các thay đổi trước khi đẩy lên**: git show <ID commit>

# [**Best (and safest) way to merge a git branch into master**](https://stackoverflow.com/questions/5601931/best-and-safest-way-to-merge-a-git-branch-into-master)

git checkout master

git pull origin master

git merge test

git push origin master

Say your bug fix branch is called bugfix and you want to merge it into master:

git checkout master

git merge --squash bugfix

git commit

This will take all the commits from the bugfix branch, squash them into 1 commit and then merge it with your master branch.

QUY TRÌNH:

Local branch => **Push** => Remote branch => **Merge** => Remote master => **Pull** => Local master